

## MODEL FITTING SOFTWARE DEVELOPED BY JMMC FOR OPTICAL INTERFEROMETRY DATA ANALYSIS

Béchet, C.<sup>1</sup>, Tallon, M.<sup>1</sup>, Tallon-Bosc, I.<sup>1</sup> and Thiébaud, E.<sup>1</sup>

**Abstract.** The VLTI is expected to produce a large amount of interferometric observations, accessible to non-specialists. This situation led Jean-Marie Mariotti Center (JMMC) to develop a model fitting software, so that general users can understand and analyse their interferometric data. The software provides the user with a library of basic models, guidelines and advice for analysis. The fitting engine has been especially designed for this kind of optimization. This is supposed to ensure easy use of basic models and easy creation of more specific models by the user, without spoiling the fitting engine performance.

The prototype software has been successfully tested on real data.

### 1 Introduction

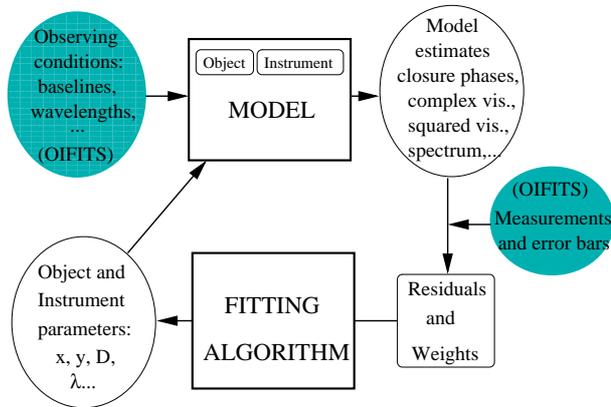
Jean-Marie Mariotti Center (JMMC) is developing a model fitting software to analyse data obtained with the VLTI or others optical interferometers. The prototype carried out at CRAL is now usable. Tallon et al. (2004) explain how precious is a model fitting software for these observations, so as to quantify dimensions of the observed objects. We briefly present the prototype contents and main improvements currently studied. Then, the first, promising, results of the software on real data are presented to enhance what else could be expected from this tool.

### 2 Software elements: MODEL function and FITTING engine

The software layout is shown in Fig. 1, highlighting two main components. On the one hand, the MODEL is aimed at computing observable quantities in the  $(u, v)$ -plane given the conditions of observation and a parameterized description of the object and the instrument. On the other hand, the FITTING engine is working directly in the  $(u, v)$ -plane. It seeks for the best parameters by maximizing the statistical consistency of the model with the observed data.

---

<sup>1</sup> CRAL UMR5574 CNRS / ENS Lyon/UCBL Observatoire de Lyon - 9, Avenue Charles André 69561 Saint Genis Laval Cedex



**Fig. 1.** Structure of the Model Fitting Software

### 2.1 A parameterized model

The MODEL part links a parameterized brightness distribution on the sky with some corresponding synthesized data on the Fourier plane. It can be seen as a function selected by the user for correctly describing the object and the instrument. A standard library of objects models has been implemented but user needs will make this package grow. The following rules for model implementation have already emerged. The model inputs are the observing conditions (extracted from OIFITS files) and a short list of parameters (until a few tenths of parameters advised). The outputs are observable quantities, such as squared visibilities, phase closures, differential visibilities or spectrum, to be compared with measured values. An implemented model must be able to initialize default values and, optionally, bounds for its parameters. It also has to be able to compute observable quantities, at least complex visibilities, but it is not in charge of computing the corresponding partial derivatives with respect to the parameters.

The existing library of objects models contains classical geometrical objects such as point-like source, uniform disk, Gaussian distribution, ring or limb-darkened distribution. These building blocks can be combined, rotated or stretched so as to generate various multiple systems. Only ideal instruments have been considered so far, which means that complex visibilities are simply the Fourier transform of the object model. This list of available models is designed to evolve and to include particular models for differential interferometry and models resulting from the combination of intensity maps. It is also possible for the model fitting software to drive an astrophysical model, on condition that inputs and outputs match. The next expectation from the users would be to take into account instrumental effects. Analytical models of realistic instruments are therefore under study.

## 2.2 A fitting engine

The fitting process consists in optimizing a “best-fit” criterion. Generally, this criterion is non-convex. Despite of these difficulties, the small number of parameters allows us to efficiently apply the Levenberg-Marquardt algorithm combined with a Trust Region method. The current algorithm is based on MINPACK routines (Moré, J. et al., 1980), which we have improved to account for bounds and to automatically compute partial derivatives by finite differences. Furthermore, the user now can keep an eye on the criterion evolution, stop or resume the process after any iteration. The outputs of the fitting procedure (best  $\chi^2$ , standard deviations, correlation matrix and confidence level) are provided so that the user can assess his/her model validity.

## 3 An example of use on real data

The software prototype has already been used to analyse on real data. The observations were part of the program “main pre-sequence stars” led by J.P. Berger, J.D. Monnier and R. Millan-Gabet on IOTA with the IONIC3 instrument (Berger et al., 2003). The first fitted model on these data was a binary system of two unresolved components. The best solution obtainable by this model yields a reduced  $\chi^2$  as bad as  $\sim 4$ , whereas a relevant model description would lead to  $\chi^2 \sim 1$ . Even for such a simple model (a binary star), the criterion is non-convex. Since the optimization algorithm will converge to any of the local minima of the criterion, the result of the fit may not be the best one. However, if the separation  $(x, y)$  of the binary is given, the criterion is convex with respect to the remaining parameter (the intensity ratio). This allows us to plot a map of the best  $\chi^2$  onto a 2-D grid of given separation as shown in Fig. 2 (right).

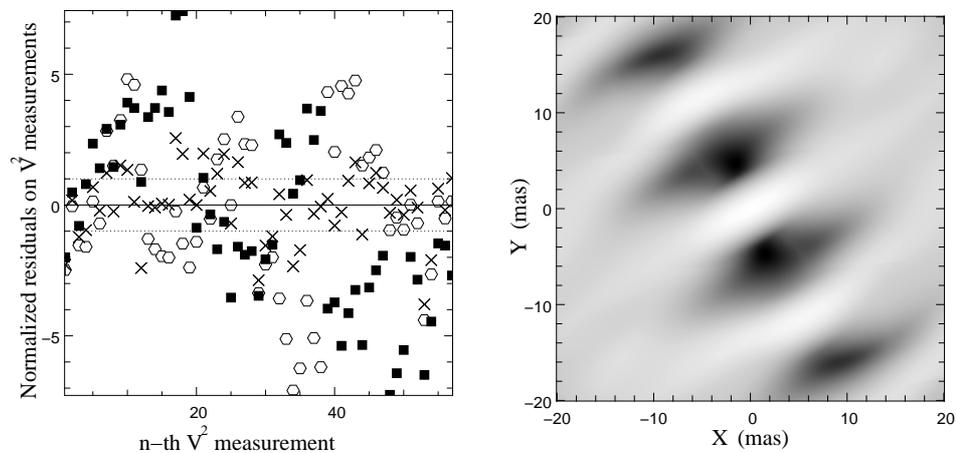
Just considering one half of the map<sup>1</sup>, the two main minima account for, at least, two characteristic lengths in this object. We therefore fit a 3-component model to the data and obtained a reduced  $\chi^2$  of 1.65. On the left part of Fig. 2, for each squared visibility measurement, the differences from the model estimates normalized by the error bars are plotted. The triple system leads to smaller absolute residuals and since then, the observed object has been confirmed to be triple. This first study on real data was all the more interesting as it helped finding out useful tools derived from the model fitting software to unveil the true nature of the object.

## 4 Perspectives

The prototype is actually entirely coded in Yorick language and will be translated into C. Till then, the Yorick software is operational to analyse real data. We expect the use of the software by various astronomers to enrich the model library.

---

<sup>1</sup>The symmetry of Fig. 2 is due to the symmetry of the problem when the brightest component is not constraint to lie at  $(0, 0)$ .



**Fig. 2.** Model fitting on real data. **Left:** Normalized residuals on  $V^2$  measurements:  $(V_{\text{model}}^2 - V_{\text{meas}}^2) / \sqrt{\text{Var}(V_{\text{meas}}^2)}$ . Squares and circles correspond to the two best fitted binary systems (2 local minima). Crosses represent the residuals for the triple system. **Right:** On a grid of given separations  $(x, y)$ , the intensity ratio of a binary system has been fitted. Final  $\log(\chi^2)$  are mapped. Dark spots pinpoint local minima.

### Acknowledgements

We especially thank J.P. Berger for having provided us with real data.

### References

- Berger, J.P., Haguenaer, P., et al., Feb. 2003, An Integrated-Optics 3-Way Beam Combiner for IOTA, ed. Wesley A. Traub, Proceedings of the SPIE, 4838, pp. 1099-1106
- Moré, J., Garbow, B., & Hillstom, K., 1980, MINPACK
- Tallon, M., Tallon-Bosc, I., et al., 2004, JMMC Model Fitting Software For Optical Interferometry, SF2A 2004, EDP Sciences